Write a python program to compute Central Tendency Measures: Mean, Median, Mode. Measure of Dispersion: Variance, Standard Deviation.

There are three main measures of central tendency which can be calculated using the methods in python library.

- Mean :- It is the average value of data which is a division of sum of values with number of values

Program :-
```
import numpy
Speed = [99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85, 86]
X = numpy.mean(Speed)
print(x)
```
Output :- 89.76

- Median :- It is the middle value in distribution when the values are arranged in ascending or descending order.

Program :-
```
import numpy
Speed = [99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85, 86]
X = numpy.median(Speed)
print(x)
```
Output :- 87.0

Program :- import pandas as pd

```
# Create a Dictionary of Series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith',
                       'Jack','Lee','Chanchal','Gasper','Nortya','Andrew']),
     'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),
     'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,
                         2.98,4.86,4.10,3.65])}

# Create a DataFrame
df = pd.DataFrame(d)
print("Mean value in the 'Distribution")
print(df.mean())
print('* * * * * * * * * * * * * * * *')
print("Median values in the Distribution")
print(df.median())
```

- Mode - The mode value is the value that appears the most number of times.

Program:-

```
from Scipy import stats
Speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]
X = Stats.mode(Speed)
print(x)
```

output :- 86

Program :- import Statistics

```
Set1 = [1, 2, 3, 3, 4, 4, 4, 5, 5, 6]
Print ("Mode :" % (statistics.mode(Set1)))
```

Output :- 4

Variance - Variance is a statistical measurement of the spread between numbers in a data set

$$\sigma^2 = \frac{\sum (x - mean)^2}{N}$$

Program :- import statistics

```
Sample = [1, 2, 3, 4, 5]
print ("Standard Deviation of Sample is %s" % (statistics.stdev
                                                (Sample)))
print ("Variance of Sample is %s" % (statistics.variance (Sample)))
```

Program :- import numpy

```
Sample = [1, 2, 3, 4, 5]
diff = [(x - numpy.mean(Sample)) ** 2 for x in Sample]
print sum(diff) / len(Sample)
```

Standard Deviation - It is the square root of the variance.

$$S.D = \sqrt{\frac{\Sigma(x - mean)^2}{N}}$$

Program:-

```
import numpy.

Speed = [86, 87, 88, 86, 87, 85, 86]
x = numpy.std (Speed)
print(x)
```

Output:- 0.9035079029052513

4/8/24

GROUP OF INSTITUTIONS
EXPLORE TO EVENT

Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy.

Statistics:- Python has a built-in module that you can use to calculate mathematical statistics of numeric data.

Program :-1

```
import statistics
data = [1,2,3,4,5,6,7,8,9,10]
mean = statistics.mean(data)
median = statistics.median(data)
Print ("Mean :", mean)
Print ("Median :", median)
```

Output :- Mean : 5.5
Median : 5.5

Program :-2

```
import statistics
data = [1,2,3,4,5,6,7,8,9,10]
print ("Standard Deviation : %", % (Statistics.stddev(data)).
```

Numpy :- It is used for numerical operation and support large, multi-dimensional array and matrices, along with a collection of mathematical functions to operate on these arrays

Program :-

```
import numpy as np
arr = np.array([1,2,3])
print(arr)
```

output :- [1,2,3]

Program :-

```
import numpy as np
arr = np.array([1,2,3])
print(np.mean(arr))
```

output :- 2.0

Math :- It provides mathematical function for basic operation, such as trignometry.

Program :-

```
import math
print(math.sqrt(16))
```

output :- 4.0

Program :-

```
import math
print(math.factorial(5))
```

output :- 120

SciPy:- It builds on NumPy and provides additional functionality for scientific and functional computing, including optimization, integration, and Eigen value problems.

Program:-

```
from scipy.integrate import quad
result, error = quad(lambda x: x**2,0,1)
print(result)
```

output - 0.3333333333337

Program:-

```
from scipy.optimize import minimize

result = minimize(lambda x: x**2,0)
print(result.x)
```

output : [0.]

21/3/24.

rite a Python Program to implement Simple linear Regression

ogram: -

```
import pandas as pd.
import numpy as np
import matplotlib.pyplot as plt.

dataFrame = pd.read_csv ('Age_Income.csv')

age = dataFrame ['Age']
income = dataFrame ['Income']
num = np.size (age)
mean_age = np.mean (age).
mean_income = np.mean (income).

CD_ageincome = np.Sum (income *age) - num * mean_income * mean_age.
CD_ageage = np.Sum (age * age) - num * mean_age * mean_age.

b1 = CD_ageincome / CD_ageage.
b0 = mean_income - b1 * mean_age.
print ("Estimated Coefficients:")
print (" b0 = ", b0, "\n b1 = ", b1)

plt.scatter (age, income, color = "b", marker = "2")
response_vec = b0 + b1 * age.
```

```
plt.plot(age, response_vec, color = "r")
plt.xlabel('Age')
plt.ylabel('Income')
plt.show()
```

Dataset:-

Age, Income
25, 25000
23, 22000
24, 26000
28, 29000
34, 38600
32, 36500
42, 41000
55, 81000
45, 47500

Output:-

Estimated Coefficients.

$b_0 = -14560.450160$
$b_1 = 1550.792374827433$

Study of Python Libraries for ML application such as Pandas and Matplotlib?

Pandas :- It is primarily used for data manipulation and analysis. It provides data structure like Data Frames that make it easily to handle and analyze large datasets.

Program :-
```
import pandas as pd.
import numpy.

data = { 'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35] }
df = pd.DataFrame(data)
df.to_csv('people.csv', index = False)
df_read = pd.read_csv('people.csv')
print("Data Frame read from csv:")
print(df_read).
```

Output :-

Data Frame read from csv:

	Name	Age
0	Alice	25
1	Bob	30
2	Charlie	35

Matplotlib:- It is a plotting library used to create static, interactive, and animated visualizations in Python.

Program:-

```
import matplotlib.pyplot as plt

X = [1,2,3,4,5]
Y = [2,3,5,7,11]
plt.plot(X,y, marker = 'o')
plt.xlabel('x axis')
plt.ylabel('Y axis')
plt.title('Simple Line Plot')
plt.show()
```

CMR GROUP OF INSTITUTIONS EXPLORE TO INVENT

Implementation of multiple linear Regression for House Price Prediction using sklearn?

Program:-

```
import pandas as pd.
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_section import train_test_split
from sklearn.linear_model import LinearRegression,
from sklearn.metrics import mean_squared_error, mean_absolute_error

df = pd.read_csv('Real-estate1.csv')
df.drop('No', inplace = True, axis = 1)

print(df.head())
print(df.columns)

sns.scatterplot(x = 'X4 number of convenience stores',
                y = 'Y house price of unit area', data = df)

x = df.drop['Y house price of unit area', axis = 1]
y = df['Y house price of unit area'].

print(x)
print(y)
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3,
                                   random_state=101)

model = LinearRegression()
model.fit(X_train, y_train)

prediction = model.predict(X_test)

print('mean_squared_error:', mean_squared_error(y_test, predictions))
print('mean_absolute_error:', mean_absolute_error(y_test, prediction))
```

Output:-

	X1 transaction date	X2 house age	...	X6 longitude	Y house price of unit area
0	2012.917	...		121.54024	37.9
1	2012.917	...		121.53951	42.2
2	2013.583	...		121.54391	47.3
3	2013.500	...		121.54391	54.8
4	2012.833	...		121.54243	43.1

[5 rows x 7 columns]

Index(['X1 transaction date', 'X2 house age', 'X3 distance to the nearest MRT station', 'X4 number of convenience stores', 'X5 latitude', 'X6 longitude', 'Y house price of unit area'], dtype='object')

	X1 transaction date	X2 house age	.....	X5 latitude	X6 longitude
0	2012.917	32.0	....	24.98298	121.54024
1	2012.917	19.5	....	24.98034	121.53951
2	2013.583	13.3	....	24.98746	121.54391
⋮	⋮	⋮		⋮	⋮
411	2013.250	18.8	.....	24.94293	121.53983
412	2013.000	8.1	.....	24.96604	121.54067
413	2013.500	6.5	.....	24.97433	121.54310

[414 rows x 6 columns]

0	37.9
1	42.3
3	42.3
⋮	⋮
412	40.6
413	63.9

Name: Y house price of unit area, length = 414,
dtype = float64

Mean Squared Error : 46.21179783493614

Mean absolute Error : 5.392293684756542.